Duplicate Records Detection and Prioritization: A Case Study for a U.S. County

By Andrea Rozario
Of Rutgers, the State University of New Jersey
for the
36th WCARS
June 2, 2016
Sao Paulo, BR
Continuous Auditing

- Continuous auditing entails the real-time monitoring and analysis of the entire population of records (Vasarhelyi and Halper 1991)
- Premise of this methodology is based on the concept of audit-by-exception where deviations (e.g. control variances) are flagged as alerts and forwarded to the responsible parties (e.g. management, internal auditors, business owners) for investigation
- There is an increasing trend to follow an audit-by-exception approach
- Important to maintain a high level of quality of data in order to rely on the results of such approach
Why is the detection of duplicate records important?

• Business and governmental entities generate a substantial amount of data every day

• This data is used to perform analyses that can support decision making:
  – Using prior year purchasing data as a baseline to create an expenditure budget
  – Assuring the quality of the CAFR (Comprehensive Annual Financial Report)

• Important to ensure the quality of the data that is generated by an entity’s relational database

• Shortage of studies that address the problem of duplicate records in the governmental accounting literature

• CA literature is rich with studies that propose statistical and machine learning techniques to identify exceptions, but the results of duplicate records detection are usually too many (Dull et al., 2006; Kogan et al., 1999)
What is the issue with identifying too many duplicates?
Solution to duplicate record detection problem

- How can we devise a methodology to rank the detected duplicates in order to enable the human users to focus their attention on the more suspicious cases?
Duplicate records

Costly Problem

Causes:
- Different formats, structures or schema of databases
- Lack of a global or unique identifier
- Human factors (data entry, lack of constraints, intentional)

Detection Methods:
1. Exact matching:
   Records are identical
2. Fuzzy (near-identical) matching (Weis et.al., 2008):
   - Records have similar values for certain relevant fields
   - Causes: data entry errors, different value formats, etc.
     E.g. 10/21/10 vs. October 21, 2010
   - Classified as duplicates based on a threshold and some similarity criteria

<table>
<thead>
<tr>
<th>Vendor Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>J.B. Smith</td>
<td>1 Washington Park</td>
</tr>
<tr>
<td>J. Smith</td>
<td>1 Washington Park</td>
</tr>
<tr>
<td>John Smith</td>
<td>1 Washington Park Ave</td>
</tr>
<tr>
<td>John Smith</td>
<td>1 Washington Park Avenue</td>
</tr>
</tbody>
</table>
Duplicate Detection Process

**Generalized framework** (Weis & Neumann, 2005):

- **Phase 1: Candidate definition** *(offline)*
  - Determine which objects to compare

- **Phase 2: Duplicate definition** *(offline)*
  - Determine criteria (description + similarity measure) to use in order to consider actual duplicates

- **Phase 3: Actual duplicate detection**
  - Specifying how to detect duplicates candidates and find which ones are true duplicates

<table>
<thead>
<tr>
<th>Record</th>
<th>Vendor Name</th>
<th>Address</th>
<th>Age</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John Smith</td>
<td>1 Washington Park</td>
<td>32 yrs</td>
<td>973-123-4567</td>
</tr>
<tr>
<td>2</td>
<td>J.B. Smith</td>
<td>1 Washington Park</td>
<td>32 years</td>
<td>1-973-123-4567</td>
</tr>
<tr>
<td>3</td>
<td>J. Smith</td>
<td>1 Washington Park</td>
<td>32 years</td>
<td>(973)1234567</td>
</tr>
<tr>
<td>4</td>
<td>John Smith</td>
<td>1 Washington Park Ave</td>
<td>32 years</td>
<td>+1-973-123-4567</td>
</tr>
<tr>
<td>5</td>
<td>John Smith</td>
<td>1 Washington Park Avenue</td>
<td>32 yrs</td>
<td>+19731234567</td>
</tr>
</tbody>
</table>
Data

**Data Description**
1 file: (August 2011 – June 2015)

- **Dataset:** information on payments to various vendors; 473,000 records, 230 variables

**Software & Algorithm used**
Excel (data cleaning and preparation)
IDEA (duplicates detection)
Algorithm: 3-way match (Payee + Invoice Date + Invoice Amount)
   - Additional variable: Invoice number
Algorithms and Findings

**Dataset**
- (Date, Amount, Vendor) yielded 83,000 candidates
- (Date, Amount, Vendor, Invoice ID) yielded 8,000 candidates
Duplicate Candidates Prioritization

- Large numbers of candidates
- Use a set of criteria to differentiate (rank) between them
- Simply adding a new variable to the algorithm proved suboptimal

**Proposed prioritization based on a Composite Score:**

\[ CS_i = \sum W_{icr_j} \]

Where \( CS_i \) is the Composite Score of the set of duplicate candidates \( i \)

\( W_{icr_j} \) is the weight of Criterion \( j \) when applied to the set of duplicate candidates \( i \)

**Proposed set of criteria:**

Materiality, missing values, count of similar candidates, frequency per user, frequency per vendor, duplicate invoice number
Prioritization Criteria

- **Materiality:** \( W_{i,Materiality} = \frac{Amt_i}{\sum Amt_i} \)

- **Missing values:** \( W_{i,MissValue} = \)
  \[
  \begin{cases} 
  1/(\sum \text{Count}_i), & \text{if the set of duplicate candidates } i \text{ does not have missing values} \\
  0, & \text{Otherwise}
  \end{cases}
  \]

- **Count of similar candidates:** \( W_{i,Count} = \frac{\text{Count}_i}{\sum \text{Count}_i} \)

- **Frequency per user:** \( W_{i,FreqUser} = \frac{\text{Count}_{Uji}}{\sum \text{Count}_i} \)

- **Frequency per vendor:** \( W_{i,FreqVndr} = \frac{\text{Count}_{Vji}}{\sum \text{Count}_i} \)

- **Duplicate invoice number:** \( W_{i,InvID} = \)
  \[
  \begin{cases} 
  1/(\sum \text{Count}_i), & \text{if the Invoice ID is the same for the candidates} \\
  0, & \text{Otherwise}
  \end{cases}
  \]
**Prioritization Example**

For Record 1001 we calculate the following weights:

- \( W_{1001_{\text{Materiality}}} = \frac{\text{Amt}_{1001}}{\sum \text{Amt}_i} = \frac{268.55}{9205.35} = 0.0292 \)
- \( W_{1001_{\text{MissValue}}} = \frac{1}{\sum \text{Count}_i} = \frac{1}{7} = 0.1429 \) (as there are no missing values causing it to be a duplicate candidate)
- \( W_{1001_{\text{Count}}} = \frac{\text{Count}_{1001}}{\sum \text{Count}_i} = \frac{3}{7} = 0.4286 \)
- \( W_{1001_{\text{FreqUser}}} = \frac{\text{Count}_{U_{ji}}}{\sum \text{Count}_i} = \frac{5}{7} = 0.7143 \)
- \( W_{1001_{\text{FreqVndr}}} = \frac{\text{Count}_{V_{ji}}}{\sum \text{Count}_i} = \frac{3}{7} = 0.4286 \)
- \( W_{1001_{\text{InvID}}} = \frac{1}{\sum \text{Count}_i} = \frac{1}{7} = 0.1429 \) (Invoice ID are the same)

\[
\text{CS}_{1001} = 1.8863
\]
## Ranking of the example

Composite Scores of all the duplicate candidates in the example:

<table>
<thead>
<tr>
<th>Record #</th>
<th>Score - Materiality</th>
<th>Score - Missing Values</th>
<th>Score - Count</th>
<th>Score - Frequency by User</th>
<th>Score - Frequency by Vendor</th>
<th>Score - Invoice ID</th>
<th>Composite Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>0.0292</td>
<td>0.1429</td>
<td>0.4286</td>
<td>0.7143</td>
<td>0.4286</td>
<td>0.1429</td>
<td>1.8863</td>
</tr>
<tr>
<td>2034</td>
<td>0.0292</td>
<td>0.1429</td>
<td>0.4286</td>
<td>0.7143</td>
<td>0.4286</td>
<td>0.1429</td>
<td>1.8863</td>
</tr>
<tr>
<td>9418</td>
<td>0.0292</td>
<td>0.1429</td>
<td>0.4286</td>
<td>0.7143</td>
<td>0.4286</td>
<td>0.1429</td>
<td>1.8863</td>
</tr>
<tr>
<td>7430</td>
<td>0.4475</td>
<td>0.0000</td>
<td>0.2857</td>
<td>0.2857</td>
<td>0.5714</td>
<td>0.0000</td>
<td>1.5904</td>
</tr>
<tr>
<td>6159</td>
<td>0.4475</td>
<td>0.0000</td>
<td>0.2857</td>
<td>0.2857</td>
<td>0.5714</td>
<td>0.0000</td>
<td>1.5904</td>
</tr>
<tr>
<td>8332</td>
<td>0.0087</td>
<td>0.1429</td>
<td>0.2857</td>
<td>0.7143</td>
<td>0.5714</td>
<td>0.0000</td>
<td>1.7230</td>
</tr>
<tr>
<td>4723</td>
<td>0.0087</td>
<td>0.1429</td>
<td>0.2857</td>
<td>0.7143</td>
<td>0.5714</td>
<td>0.0000</td>
<td>1.7230</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>6</td>
</tr>
</tbody>
</table>
Conclusion

• Given the recent emphasis on transparency and accountability of government funds, it is important to ensure the data is accurate and reliable
• In this study, we detected duplicate candidates for a U.S. county and proposed a prioritization framework to rank these candidates
• Next step: Apply the prioritization framework to the government data and refine the framework as we obtain feedback